On using wide-area inter-domain OpenFlow paths

S. Tepsuporn*, M. Veeraraghavan*, B. Cashman†, A. J. Ragusa‡, L. Fowler‡, Chin Guok§, T. Lehman and X. Yang¶
* University of Virginia, Charlottesville, VA 22904–4743, {spt9np,mvee}@virginia.edu
† Internet2, Ann Arbor, MI 48104, bsc@internet2.edu
‡ Indiana University, Bloomington, IN 47405, {aragusa,luke}@grnoc.iu.edu
§ Energy Sciences Network (ESnet), Lawrence Berkeley National Laboratory, Berkeley, CA 94720, chin@es.net
¶ University of Maryland, College Park, MD 20742, {tlehman,maxyang}@umd.edu

***Abstract*—This paper describes our experience with a new wide-area inter-domain dynamic path-based networking service. OpenFlow switches with two controllers, OESS to perform intra-domain topology discovery and path provisioning actions, and OSCARS for inter-domain provisioning, were deployed in several university campuses, and regional and core research-and-education networks (RENs). As part of this work, we configured the equipment and controllers at 8 university sites, used OESS and OSCARS to dynamically set up inter-university VLAN Layer-2 (L2) paths. In documenting our experience, we describe the administrative steps necessary for configuring controllers in this centralized software-defined network model, offer methods for debugging L2 path setup failures, and outline the VLAN configuration steps needed at end hosts to run applications without modifications over L2 paths. We ran file-transfer tests on these paths and learned that a core REN had deployed a policing mechanism that directed out-of-profile packets to a separate queue, which caused a significant percent of retransmissions due to out-of-sequence packet delivery. These retransmissions were reduced by limiting the average and peak rates at the sending host to the path rate by using the Linux `tc` utility, but TCP congestion window drops still occurred. However, when we reduced the average and peak `tc` sending rates to slightly below the path rate, then throughput reached close to the path rate.**

***Keywords*—OpenFlow, SDN, path-based networking, VLAN, WAN**

I. Introduction

The primary inter-domain networking service offered today on the Internet is IP. IP is a datagram service, which means routers have destination based forwarding tables. In contrast, path-based networking (also known as virtual circuit (VC) networking) services require a setup phase in which path-based forwarding table entries are created at switches before user data transfer. Ethernet VLAN technology, and the introduction of OpenFlow and Software Defined Network (SDN) controllers, offer a new means to enable path-based networking.

The scientific high-performance computing and networking community has identified high-speed file transfers as a motivating application for path-based networks. Several papers have described how even with more aggressive congestion control algorithms such as HTCP [1], packet losses can cause a significant reduction in throughput on high bandwidth-delay product paths. More importantly, file transfers are part of scientific workflows that require co-scheduling of different types of resources, such as compute, storage, visualization, and scientific instruments. Such co-scheduling requires file transfers across wide-area networks to have predictable delays. It is for these reasons that the Research-and-Education Network (REN) community has added a dynamic Layer-2 path-based service to complement their best-effort IP service.

The *objective* of this work was to use and evaluate the current deployment of these inter-domain wide-area path based services. Specifically, the paper describes our experiences with the control-plane software used to dynamically establish and tear down paths, and provides results from data-plane experiments for file transfers across these paths.

The *novelty* and *significance* of this work lies in its use of a wide-area multi-domain deployment of OpenFlow switches with control-plane software, and hosts with data-plane software to run applications on path-based services. While prior work has reported on data-plane experiments for large file transfers across virtual circuits on single-organization owned WAN testbeds [2], this paper describes a concerted multi-year, multi-organization effort to (i) develop control-plane software for multi-domain Layer-2 (L2) path reservation and provisioning, (ii) deploy and configure OpenFlow switches, hosts and the control-plane software at various university campuses, regional and backbone networks, and (iii) experiment with applications for this path-based networking service. Our *contributions* include (i) detailed reporting on our experiences in working with the controllers and debugging failed setup attempts in a multi-domain context, (ii) methods for configuring end-hosts for IP-based applications to operate across L2 paths, and (iii) methods to configure sending rates to handle the mismatch between TCP congestion control and providers' policing mechanisms for path-based networks.

Section II provides background information on the multi-domain deployment of path-based networking service, and reviews related work. Section III describes the control-plane software used for multi-domain path reservation, provisioning and release. Section IV describes our experiences with configuring and using the dynamic path-based service. Section V describes our data-plane experiments with file transfers across wide-area multi-domain paths, and lessons learned. The paper is concluded in Section VI.

II. Background and Related Work

Background

Fig. 1 shows two university domains with Dynamic Network System (DYNES) deployments, two regionals, one of which offers dynamic path-networking service and therefore has an Open Exchange Software Suite (OESS)[1] and On-Demand Secure Circuits and Advance Reservation System (OSCARS)[2], while the other does not, and three backbone

[1] http://globalnoc.iu.edu/sdn/oess.html
[2] http://www.es.net/services/oscars/

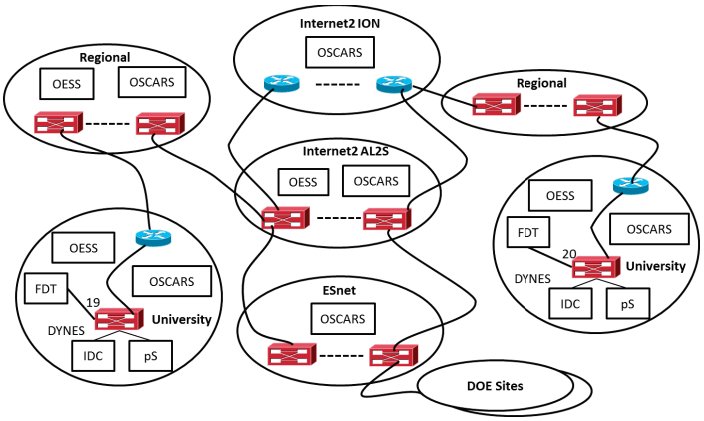

Fig. 1: A multi-domain dynamic path-based networking service; File Data Transfer (FDT) server; Inter-Domain Controller (IDC); perfSONAR (pS) host; Open Exchange Software Suite (OESS); On-Demand Secure Circuits and Advance Reservation System (OSCARS); Interoperable On-demand Network (ION); Advanced Layer 2 Service (AL2S)

networks, Internet2's AL2S and ION[3], and ESnet[4]. ESnet serves DOE sites, while universities typically connect to regionals, and through their regionals, access Internet2's services. Regionals that have not deployed OESS and OSCARS still support static path-based networking service.

Related work Multiple projects were started during the 2003-2006 timeframe to address the issue of dynamic provisioning of path-based networking service within the REN community. This work was motivated by a desire to develop alternatives and enhancements to best-effort IP service for large-bandwidth science data flows. DOE funded the OSCARS [3], UltraScience Net (USN) [4], TeraPaths [5] and Lambda Station [6] projects. NSF funded the Circuit-switched High-speed End-to-End Transport ArcHitecture (CHEETAH) [7] and Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON) [8] projects. Contributions from these projects, and international collaborations with the Europe's GEANT[5], Japan's JGN-X[6], Canada's CANARIE[7], and Brazil's RNP[8], resulted in the OESS and OSCARS controllers shown in Fig. 1. The objective was to provide both packet services, e.g., best-effort IP, and dynamic path-based networking service over a common infrastructure with the control-plane intelligence. This objective has been realized in DOE's Energy Sciences Network (ESnet), Internet2 on their ION and AL2S networks, and increasingly on university campuses and regionals as illustrated in Fig. 1.

The NSF Global Environment for Network Innovation (GENI) program has also fostered several interesting research projects that look for network innovations from the inter-domain end-to-end perspective. The GENI infrastructure spans many domains, a.k.a. aggregates, to support experiments at scale. This led to the introduction of the GENI Network Stitching Architecture[9] that helps create multi-tenant network virtualizations across distributed domains/aggregates. Projects such as ProtoGENI[10] and ExoGENI[11] have developed proprietary multi-domain solutions to allocate internal network resources across their global footprints.

III. Control Plane Software

Fig. 1 shows two control-plane software systems, named OESS and OSCARS, to support dynamic path-based networking service in university campus, regional and core networks. As OESS is an OpenFlow controller, and ESnet and Internet2 ION do not have OpenFlow switches, they deploy only OSCARS. Both OSCARS and OESS offer users a Web Browser User Interface (UI) and a programmatic Web Service Interface for applications. In this section, we briefly review the functionality offered by these control-plane modules.

A. OSCARS

We review the overall OSCARS framework, describe how trust/peering relationships are established between neighboring OSCARS, and how topology is discovered before presenting how OSCARS reserves resources, provisions and releases paths (which is its main role). We end with a short review of path computation, which is executed during resource reservation [3].

Architecture/framework The OSCARS software consists of 11 modules that have distinct functions such as authentication, authorization, path finding, messaging, hardware mediation, and process coordination. Today, OSCARS supports inter-domain L2 VCs using both the Inter-Domain Controller Protocol (IDCP) [9] and Network Service Interface Connection Services (NSI CS) version 2.0 [10] protocols. The authorization (i.e., policy enforcement) of guaranteed-bandwidth reservation requests are domain specific, and can be enforced using the policy path computation modules within the OSCARS v0.6 Path Computation Engine (PCE) framework. In practice, in the R&E community, not all path-networking service deployments have policing, as will be seen later in Section V.

Trust/peering relationships The current trust model for inter-domain dynamic VCs is based on transitive peer-to-peer authentication and authorization. This work-flow mimics the telecommunication industry model, which does not require downstream providers to know anything about the originating caller.

Topology discovery Each domain is responsible for discovering and pushing its topology to the perfSONAR Topology Service (pS-TS)[12]. The distributed pS-TS maintains global topology information, and OSCARS servers can pull the latest information from pS-TS as needed in real-time. Topology information must be formatted in the Open Grid Forum Network Markup Language (NML) [11] or NM-Control Plane[13] schemas to support the NSI CS v2.0, and the IDCP protocol, respectively.

[3] http://www.internet2.edu/
[4] http://www.es.net/
[5] http://services.geant.net/bod/Pages/Home.aspx
[6] http://www.jgn.nict.go.jp/english/index.html
[7] http://www.canarie.ca/en/home
[8] http://www.rnp.br/en

[9] http://groups.geni.net/geni/wiki/GeniNetworkStitching
[10] http://www.protogeni.net
[11] http://www.exogeni.net
[12] http://psps.perfsonar.net/
[13] http://www.controlplane.net/idcp-v1.1/nmtopo_ctrlplane.rnc

Inter-domain VC reservation, provisioning, and release When the OSCARS server in a domain receives an inter-domain VC reservation request, it reserves resources within its own domain and sends a `createReservation` message[14] with endpoints, rate, start time (advance-reservation support) and duration, to the OSCARS in the next domain, which is selected based on the computed path. The procedure is executed in a daisy-chain until the OSCARS of the last domain on the end-to-end path is reached. If successful, `Confirmation` events are sent from one domain's OSCARS server to the next in the reverse direction. Provisioning of the VC occurs either automatically or upon receiving a `createPath` message from the user just before the reservation start-time. This procedure also uses a daisy-chain of signaling messages between OSCARS servers. Each OSCARS server communicates with the switches in its domain to provision the VC across the domain. Finally, when the reservation end-time is reached a `teardownPath` message is sent in daisy-chained mode to release the VC.

Path computation Path computation in OSCARS v0.6 is executed using "atomic" PCE modules that can be arbitrarily linked together. Each PCE module typically addresses a specific constraint and prunes the graph accordingly. For example, a bandwidth PCE would discard all links that do not have sufficient bandwidth, and a policy PCE would remove all resources that the requester is not authorized to use. While the PCE methods surveyed in [12] are for immediate-request VCs, the OSCARS PCE supports advance-reservation VCs.

B. OESS

The Open Exchange Software Suite (OESS) is an OpenFlow controller used to configure and control dynamic L2 VLAN VCs across a network of OpenFlow-enabled switches. OESS provides sub-second circuit provisioning, automatic circuit failover, per-interface permissions, and automatic per-VLAN statistics. OESS supports integration with OSCARS for inter-domain circuits.

OpenFlow path provisioning When a user wishes to provision a VLAN VC in OESS, the user must first select the endpoints (at least 2), rate, start time, duration, and optionally specify a path (with possibly a backup path) that connects all endpoints. In this context, endpoints are switch ports. Once the user has selected all of the details of their VLAN VC, the OESS UI sends the request to a Forwarding Controller. The Forwarding Controller then calculates the `OFFlowMods`, which is a specification of the OpenFlow rules required to provision the path. Each switch will receive at least 2 `OFFlowMods` (in cases of multiPoint VLANs, there can be more than 2 `OFFlowMods`). Each `OFFlowMod` is broken up into a Match and an Action.

The OpenFlow Match is applied to all packet headers, and if a packet matches all of the fields in the OpenFlow Match, all of the OpenFlow Actions for the `OFFlowMod` are then applied to the packet. OESS has implemented a specific set of OpenFlow Matches and Actions. All OESS `OFFlowMods` for a VC consist of a Match that contains the input port (IN_PORT) and input VLAN ID (DL_VLAN) fields. The Actions consist of SET_VLAN_ID and OUTPUT (to a port) actions. In some cases the STRIP_VLAN action is also used for (untagged circuits).

Topology discovery OESS learns the topology for its domain, through a protocol similar to Link Layer Discovery Protocol (LLDP), called OpenFlow Discovery Protocol (OFDP)[15]. OFDP functions by having the controller generate a packet and send the packet out on every interface of an OpenFlow switch using the `OFPacketOut` mechanism. The packet that is sent out on each interface is tagged with the `DataPathID` (a unique identifier for each OpenFlow Switch that is usually based on a management MAC address) and number of the interface on which the packet was sent. A rule is configured on all switches to "punt" these topology-discovery packets to the controller through an `OFPacketIn` event. The `OFPacketIn` event sends the packet that arrived at the switch along with the port and `DataPathID` of the switch that received the packet. When this procedure occurs in both directions, an adjacency is detected and OESS creates a link between the 2 devices on the specified ports. OESS can also detects link migrations and node insertions, allowing for the OESS to automatically move thousands of VLAN VCs with minimal human intervention.

OESS-OSCARS integration OESS integrates with OSCARS through several mechanisms. OESS automatically generates a topology file for its domain and uploads it to the OSCARS topology service. Only interfaces and VLAN IDs for which users have granted OSCARS access appear in the OSCARS topology file. When a user requests an inter-domain circuit via the OESS UI, the UI loads all topologies located in the OSCARS topology service, and presents them to the user. Once the endpoints have been selected and the user requests that a VC be provisioned, the OESS UI submits a request to OSCARS via the OSCARS SOAP API on behalf of the user. At this point the request has been turned over to OSCARS to complete its path computation, and inform the other domains of the request. When it is time for OSCARS to provision the circuit in the local domain, it contacts the Path Setup Service (PSS). When OESS is deployed in a domain, the OSCARS PSS is replaced with the OESS PSS. The OESS PSS takes the provisioning request from OSCARS and provisions an OpenFlow path as described earlier. The OESS then reports the success or failure of the provisioning procedure to OSCARS. In cases where a user request for an inter-domain VC is sent directly to OSCARS, the OESS PSS is nevertheless involved to check the validity of the VC request and to carry out the OpenFlow path provisioning.

IV. Experiences configuring and using the path-based service

As part of the NSF DYNES project [13], 40 universities and 11 regional network providers were selected for the deployment of this wide-area distributed research instrument. Each university campus DYNES equipment, as shown in Fig. 1, consists of three hosts: FDT server, IDC host, pS host, and one Ethernet switch (OpenFlow enabled in some sites). The FDT server is used for applications, the IDC host runs the control-plane software, and the pS host runs active-measurement tools for monitoring network performance. The regional sites were provided IDC and pS hosts, and an Ethernet switch. Regionals

[14] IDCP message names are used in this description, but have counterparts in NSI CSv2.

[15] OFDP: http://groups.geni.net/geni/wiki/OpenFlowDiscoveryProtocol

were not given FDTs as the latter were envisioned for use by end applications.

While some of the DYNES sites were configured for experiments as reported by Zurawski et al. [14], not all sites were configured. As part of this work, we configured and used DYNES equipment at 9 university campuses, configured OSCARS peerings for inter-domain paths, OESS for intra-domain operation, and requested and obtained static VLANs wherever needed. These sites include: (i) U. Virginia (UVA), (ii) MAX GigaPoP (MAX), (iii) Indiana University (IU), (iv) U. Wisconsin, Madison (UWisc), (v) Internet2 Lab (I2Lab), (vi) University of New Hampshire (UNH), (vii) Rutgers University, (viii) U. Colorado (CU), and (ix) U. Chicago.

A. Multi-domain dynamic path-service configuration

As can be expected with the rollout of a new networking service, organizations will slowly deploy the service one-at-a-time. The challenge lies in developing methods to support this service during this initial growth period.

Core REN providers, Internet2 and ESnet, took the lead by deploying the dynamic path-based networking service first. As described above, a few regional RENs and a few university campuses applied for and received DYNES equipment. We *first* describe how the DYNES equipment was configured at each site. *Next*, we describe the actions needed within campuses between the location of the DYNES equipment and the campus edge. *Finally*, we describe the actions required from regional RENs that did not deploy this dynamic path-based networking service.

Step 1 At each DYNES site, we logged in to the OpenFlow enabled switch, configured the IP address of the IDC host on which the OESS (OpenFlow controller) is being run, and added the set of ports to be controlled by the OESS into the switch's OpenFlow instance (only one instance is used). The OpenFlow switch models used by the DYNES project support *hybrid-switch mode* in which OpenFlow controlled and traditionally configured ports can co-exist on the switch. But these switches do not support *hybrid-port mode* in which each individual port can be controlled by both the OpenFlow controller and traditional configuration methods.

The next set of operations at each DYNES site consisted of (i) initiating OESS and OSCARS on the IDC host, (ii) providing the OESS with the switch's control-port IP address, and (iii) configuring OESS and OSCARS through their Web UIs. Specifically, the OESS UI is used to set the remote-link information for the data-plane port of the peering network. For example, the UVA DYNES switch port 1 is connected (through UVA campus and regional routers) by a static VLAN VC to port et-3/0/0.0 on the Internet2 AL2S switch in Ashburn, VA. The remote-link information entered into UVA DYNES OESS identifies the peering domain, node, and port. The counterpart action was performed at Internet2's OESS for the UVA DYNES switch remote link. This remote-link information was provided manually to Internet2. The OESS UI is also used to configure the set of allowed VLANs on each port of the DYNES switch. UVA DYNES OSCARS needed to be configured with a server certificate, and the certificate owner and issuer information needed to be manually communicated to Internet2's administrator for configuration of Internet2's AL2S OSCARS. These certificates are used in the authentication process for inter-domain VC requests.

Step 2 Most of the involved campus networks and regional RENs support static path-based services. This allowed us to request and obtain provisioned VCs with a specified set of VLAN IDs from campus network administrators. These VCs cut across the campus switches/routers between the DYNES equipment and the campus edge router. Having the capability to establish static VLANs allows for a gradual introduction of OpenFlow switches under OESS control into campus networks.

Step 3 Similarly, we contacted regional REN administrators to obtain static VCs with specified VLAN IDs across their networks to Internet2. Again this capability of using static VCs allows for a gradual addition of dynamic path-based service by different regionals at different times.

B. Virtual circuit provisioning and testing

The OESS and OSCARS software systems were relatively stable and their Web UIs were fairly easy to navigate. We primarily used the UVA DYNES OESS Web UI to request inter-domain L2 VCs from the DYNES switch port connected to UVA's FDT server to each remote campus DYNES switch port to its FDT server. The UVA DYNES OSCARS Web UI was useful to obtain details about VC paths. We also obtained logins on Internet2's OSCARS and OESS UIs allowing us to monitor the status of successful and failed VC setup attempts. Once the DYNES sites' and Internet2 OESS and OSCARS were configured, the distributed software proved to be generally robust, and we could set up and tear down inter-domain paths dynamically.

The error reporting functionality of OSCARS could be improved. For example, we experienced circuit setup failure due to a lack of bandwidth, a requested VLAN ID already in use, or because a certificate had expired. In all these cases, while OSCARS reported a failed setup attempt, the error reporting could have provided us better information for debugging. A second issue relates to setup delay and the OSCARS approach of handling only one path setup at a time. In particular, a failed circuit setup attempt causes OSCARS to wait for a user-configured timeout interval, which is currently set to 15 min. While this solution is sufficient for low call arrival rates, a programmatic test with multiple circuit setup-and-release attempts experienced excessive delays.

Given the relative lack of L2 connectivity tools comparable to L3 tools such as `ping` and `traceroute`, we developed three methods for testing path segments to identify causes of a VC setup failure. First, we asked campus and regional REN administrators to configure a specified private IP address to a specified VLAN on the port of their domain's edge router that is connected to the next domain (toward Internet2). This allowed us to run ping to verify that static VLANs across each domain were operational. Second, we used observatory hosts located at Internet2 PoPs whose ports (with specified VLANs) were made available by Internet2 to DYNES users for L2 VC testing. This allowed us to create dynamic VCs between each campus DYNES switch and an observatory host's Internet2 router port for single campus-and-regional segment testing.

Finally, we used GRNOC's routerproxy tool[16] to observe packet counts at router ports while sending repeated pings from campus FDTs on configured VLANs to localize problems.

C. FDT access and applications

Consider the two numbered ports shown in Fig. 1. These are ports 19 and 20 on DYNES switches at two different university campuses. These ports connect to FDT servers. Once a VC is established between these two ports[17], to run an application between the corresponding FDT servers, VLANs have to be configured in these servers (using Linux `vconfig`) and IP addresses need to be assigned to the VLANs (using Linux `ifconfig`). While the end-to-end path has no IP routers, IP headers are nevertheless included/extracted at the FDT servers because of the applications' use of TCP/IP sockets. Furthermore, the private IP addresses configured for the FDT VLANs at the two ends need to belong to the same subnet.

In our usage of these VCs, as described in Section V, we manually executed these VLAN and IP address configuration commands having procured privileged authorization for the execution of these commands. But for general-purpose use of path-based networking, applications should be integrated (through shell scripts or with modifications) with a signaling-client module that issues requests for paths to OESS, handles responses, and additionally configures VLANs and IP addresses at the FDTs. Further, an end-to-end session protocol is required to exchange subnet identifier/mask information to ensure that the private IP addresses assigned to the VLANs at the two end FDTs match. Since the FDT and IDC servers have multiple Ethernet interfaces, one of which is connected to the campus IP-routed infrastructure, L3 IP service is used for all signaling messages.

D. Other challenges

In the course of one year, we experienced software upgrades to OSCARS, X.509 certificate expirations, and even network connectivity changes (regionals were moved to AL2S from ION for their Internet2 access). Each of these changes required corresponding administrative actions, sometimes in several DYNES sites and Internet2. For example, OSCARS server peerings and topology files had to be updated when a regional moved its access link from ION to AL2S.

V. Data Plane Experiments

The goal of these experiments was to measure the throughput and packet loss rate of file transfers executed across multiple campus-to-campus multi-domain paths.

A. Experimental setup and execution

We obtained logins on DYNES FDT hosts at several institutions, and used five of these hosts for data-plane experiments. We used the FDT hosts at UVA, MAX, I2Lab, IU, and UWisc (see Section IV for expanded forms). Each DYNES FDT host has two Intel Xeon E5620 four-core processors (for a total of eight cores) and 24 GiB of RAM. Table I details specific machine configurations.

Host	Distribution	Kernel	File System	Storage
UWisc	Centos 5.10	2.6.38	xfs	11 TiB
UVA	Centos 5.8	2.6.38	xfs	11 TiB
MAX	Centos 5.7	2.6.38	xfs	11 TiB
IU	Debian 7	3.2.0-4	ext4	11 TiB
I2Lab	Scientific 6.5	2.6.38	-	-

TABLE I: DYNES Host Configurations

Using the Web interface to the UVA DYNES OESS, we configured multi-domain L2 paths between the UVA DYNES FDT server and each of the FDT servers at the other campuses. The UVA-IU and UVA-MAX paths passed through the respective campuses, regionals and Internet2's AL2S. There was no policing in AL2S because the switches implement OpenFlow 1.0, which does not support QoS commands. But since the UVA access link to the regional network (MARIA) is 10 Gbps (and shared between best-effort IP and path service), we limited the rate used on these UVA-IU and UVA-MAX experiments to 4 Gbps with the Linux `tc` utility.

The UVA-UWisc and UVA-I2Lab paths passed through Internet2's AL2S and ION since UVA was connected through its regional to AL2S, and UWisc and I2Lab were connected through their regionals to ION (see Fig. 1). The regional networks' access to ION are via 10 Gbps links. Since ION supports both best-effort IP and path services, ION's OSCARS was limited to assigning up to 2 Gbps on any link for its path-based service. Given this limit, we could only obtain 50 Mbps each for the UVA-UWisc and the UVA-I2Lab paths. Furthermore, ION's OSCARS does configure policing and directs out-of-profile packets to a scavenger service queue.

The `nuttcp` tool was used for 30-sec memory-to-memory tests as it conveniently outputs throughput and the number of retransmissions every sec. We needed `sudo` access on the FDT hosts of other campuses for commands related to TCP tuning and to set `txqueuelen` (required for high-speed transfers[18]), select TCP congestion control (unless otherwise mentioned `HTCP` was used in all experiments), create VLANs on host NICs and configure IP addresses for these VLANs, configure `tc` (traffic control) for rate limiting, and open and close firewall settings.

A cron job was set up to run `nuttcp` transfers once every 30 mins for up to two days on all paths. The `tc` utility was used to limit the UVA FDT to send data at the rate of the configured path for each experiment. We implemented several shell scripts to parse the collected `nuttcp` logs, and R programs to create graphs and tables.

B. Results

Tables II and III present results from `nuttcp` tests executed on the two paths UVA-IU and UVA-MAX that pass through the AL2S network (and not ION). The `tc` column uses a short-hand notation for the choice of parameters specified

[16] http://routerproxy.grnoc.iu.edu/

[17] As mentioned in Section III-B, endpoints specified in path-reservation requests are switch ports.

[18] http://fasterdata.es.net/host-tuning/linux/

Path UVA-to-	Path rate (Gbps)	tc	Throughput (Mbps)			
			Min.	Mean	Max.	IQR
IU	4	R	2933	3856	3927	39
MAX	4	R	3695	4070	4105	27
MAX	3	R	2938	3218	3262	27
MAX	3	C	3132	3221	3248	17
MAX	3	B	3124	3221	3250	19
IU	3	B	609	2973	3132	32

TABLE II: nuttcp throughput for paths through AL2S; UVA-IU RTT = 26 ms; UVA-MAX RTT = 4.4 ms

Path UVA-to-	Path rate (Gbps)	tc	Mean packet retx rate	Mean # retx in first 2 sec	% runs w/ retx in later sec
IU	4	R	0.00075	4.8	13
MAX	4	R	0.00085	47.8	12
MAX	3	R	0.0007	58.7	4
MAX	3	C	4E-05	3.3	6
MAX	3	B	4E-05	3.5	6
IU	3	B	0.006	95	7

TABLE III: Retransmissions as reported by nuttcp (AL2S paths)

in the tc class command: (i) R: only the rate argument is set to equal the value shown in the path rate column, (ii) C: both rate and ceil arguments are set to equal the value shown in the path rate column, (iii) B: rate, ceil arguments are set as in the C case, but additionally the burst argument is set to 5000 bytes. The results in Tables II and III show that on both UVA-IU and UVA-MAX paths, the throughput is close to the path rate, and retransmission rates are low. The impact of the tc arguments is seen in the mean number of retransmissions in the first two seconds. HTCP is aggressive in increasing the congestion window size. The use of a ceiling limit for the transmission rate in the tc command lowered the number of retransmissions observed in the first two seconds (from a mean value of 58.7 to 3.3 for the UVA-MAX path as seen in rows 3 and 4 of Table III). While this mean number was higher on the UVA-IU path than on the UVA-MAX path in spite of the use of ceiling and burst limits as reported in the last row of Table III, the mean numbers of retransmissions in the first two seconds on the UVA-IU path when only rate was limited (R) and when only rate and ceiling were limited (C) were 148 and 138, respectively, both larger than the 95 value shown in the last row of Table III when rate, ceiling and burst size were limited (B). Table III also shows that the percent of runs in which retransmissions were observed in later seconds was higher with the 4 Gbps setting than with the 3 Gbps setting.

Tables IV and V present results from nuttcp tests executed on the two paths UVA-I2Lab and UVA-UWisc that traverse through the ION network. Observations from Tables IV and V are as follows. *First*, the packet retransmission rate

Path UVA-to-	tc rate (Mbps)	tc	Throughput (Mbps)			
			Min.	Mean	Max.	IQR
I2Lab	50	R	42.1	45.1	47.1	0.89
I2Lab (Reno)	50	R	25.9	37.2	41.4	2.01
UWisc	45	R	44.1	44.4	44.6	0.19
UWisc	45	C	44.1	44.4	44.6	0.17
UWisc	50	R	36.5	39.7	40.9	0.61
UWisc	50	C	37.3	39.6	41.0	0.8
UWisc	50	B	37.5	39.7	40.7	0.58

TABLE IV: nuttcp throughput for paths through ION; UVA-UWisc RTT = 24.1 ms; UVA-I2Lab RTT = 27.5 ms; Path rate = 50 Mbps

Path UVA-to-	tc rate (Mbps)	tc	Mean packet retx rate	Mean # retx in first 2 sec	% runs w/ retx in later sec
I2Lab	50	R	0.07	20.5	97
I2Lab (Reno)	50	R	0.02	19.7	30
UWisc	45	R	0.002	0.83	60
UWisc	45	C	0.0015	0.73	48
UWisc	50	R	0.15	17.8	100
UWisc	50	C	0.148	17.4	96
UWisc	50	B	0.149	17.3	96

TABLE V: Retransmissions as reported by nuttcp (ION paths)

was higher on these paths through ION than on the paths through AL2S (see Tables III and V). This is because of the policing mechanism deployed in ION. HTCP keeps increasing congestion window, and even though tc rate limiting was in place, bursts appeared at the ION ingress router at rates higher than the path rate of 50 Mbps causing out-of-profile packets to be directed to the scavenger service queue. The splitting of one TCP flow's packets between two queues caused them to arrive out of sequence, which, triggered TCP's fast retransmit procedure and led to a corresponding drop in the size of the congestion window. As Fig. 2 shows, HTCP throughput kept increasing and decreasing.

Second, a comparison of HTCP and Reno was undertaken on the UVA-I2Lab path (see first two rows of Tables IV and V). Throughput was lower with Reno as was the packet retransmission rate. The less aggressive behavior of Reno in increasing congestion window is evident from Fig. 2.

Finally, Table IV shows that setting the tc average rate limit to 45 Mbps when the path rate was set to 50 Mbps resulted in better throughput than when the tc average rate was set to equal the path rate. This finding held even when the tc peak rate was limited to the path rate and a burst size limit was set. From rows 4 and 5 of Table V, for the UVA-UWisc path, we can see that there is a 2-order difference in the mean retransmission rate (0.0015 and 0.15) between the

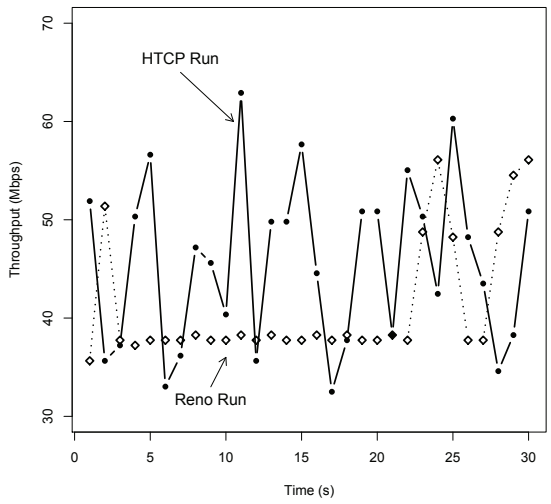

Fig. 2: Comparison of HTCP and RENO for nuttcp transfers on the UVA-I2Lab path

configurations in which tc rate limit was set to 45 Mbps and 50 Mbps. It appears that the tc rate and burst-size limiting is not very accurate.

VI. Conclusions

The research-and-education networking community has introduced a new dynamic path-based networking service with an OESS controller for a network with OpenFlow switches and an OSCARS controller for inter-domain service. This new service is complementary to best-effort IP. Our work consisted of enabling this service at 8 university campuses, configuring dynamic Layer-2 paths from the U. Virginia campus to 4 other campuses using OESS and OSCARS, and running file-transfer applications. Our contributions in the control-plane include steps required for configuring switches and controllers in the centralized software-defined network model and the potential pitfalls that can lead to path-setup failures, methods for debugging path-setup failures, and steps needed to configure end hosts to enable the use of existing file-transfer applications without modifications. Our data-plane contributions include observations on the impact of providers' policing mechanisms on TCP throughput (by sending out-of-profile packets to a different queue, segments arrive at the receiver out-of-sequence triggering fast retransmit), and methods for handling the mismatch between TCP's congestion control schemes and the policing mechanisms. We found that setting the peak rate limit with the Linux tc facility at the sending host to slightly lower than the path rate was an effective way of avoiding losses and out-of-sequence segment delivery.

Acknowledgment

The authors thank Ezra Kissel (Indiana U), Dale Carder and Jerry Robaidek (U. Wisconsin), Ivan Seskar and Steve Decker (Rutgers U), R. D. Russell and P. MacArthur (U. New Hampshire), Conan Moore (U.Colorado), and Ryan Harden (U. Chicago), Ron Withers (U. Virginia), John Lawson (MARIA), Eric Boyd (Internet2), Ben Nelson (GRNOC), and all the regional REN providers for their support. This work was supported by NSF grants CNS-1116081, OCI-1127340, ACI-1340910, and CNS-1405171, ACI-0958998, and DOE grant DE-SC0007341.

References

[1] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long-distance networks," in *Protocols for Fast Long Distance Networks Workshop (PFLDnet)*, Feb. 16-17, 2004.

[2] E. Kissel, M. Swany, B. Tierney, and E. Pouyoul, "Efficient wide area data transfer protocols for 100 Gbps networks and beyond," in *Proceedings of the Third International Workshop on Network-Aware Data Management*, ser. NDM '13. New York, NY, USA: ACM, 2013, pp. 3:1–3:10. [Online]. Available: http://doi.acm.org/10.1145/2534695.2534699

[3] C. Guok, D. Robertson, M. Thompson, J. Lee, B. Tierney, and W. Johnston, "Intra and interdomain circuit provisioning using the OSCARS reservation system," in *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, Oct 2006, pp. 1–8.

[4] N. S. V. Rao, W. Wing, S. Carter, and Q. Wu, "Ultrascience net: network testbed for large-scale science applications," *Communications Magazine, IEEE*, vol. 43, no. 11, pp. S12–S17, Nov 2005.

[5] D. Katramatos, D. Yu, B. Gibbard, and S. McKee, "The TeraPaths testbed: Exploring end-to-end network QoS," in *Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on*, May 2007, pp. 1–7.

[6] A. Bobyshev, M. Crawford, P. DeMar, V. Grigaliunas, M. Grigoriev, A. Moibenko, D. Petravick, and R. Rechenmacher, "Lambda station: On-demand flow based routing for data intensive grid applications over multitopology networks," in *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, Oct 2006, pp. 1–9.

[7] X. Zheng, M. Veeraraghavan, N. S. V. Rao, Q. Wu, and M. Zhu, "CHEETAH: Circuit-switched High-speed End-to-End Transport ArcHitecture testbed," *Communications Magazine, IEEE*, vol. 43, no. 8, pp. s11–s17, Aug 2005.

[8] X. Yang, C. Tracy, J. Sobieski, and T. Lehman, "GMPLS-based dynamic provisioning and traffic engineering of high-capacity ethernet circuits in hybrid optical/packet networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–5.

[9] Open Grid Forum (OFG) Network Service Interface Working Group (NSI-WG), "Inter-Domain Controller Protocol (IDCP) Specification," Oct. 24, 2010, http://www.ggf.org/documents/GFD.170.pdf.

[10] G. Roberts, T. Kudoh, I. Monga, J. Sobieski, J. MacAuley, and C.Guok, "NSI Connection Service v2.0," 2014, OGF GFD.212.

[11] J. vd. Ham, F. Dijkstra, R. Lapacz, and J. Zurawski, "Network Markup Language Base Schema version 1," 2013, OGF GFD.206.

[12] F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo, and P. Castoldi, "A survey on the path computation element (PCE) architecture," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 1819–1841, Fourth 2013.

[13] J. Zurawski, R. Ball, A. Barczyk, M. Binkley, J. Boote, E. Boyd, A. Brown, R. Brown, T. Lehman, S. McKee, B. Meekhof, A. Mughal, H. Newman, S. Rozsa, P. Sheldon, A. Tackett, R. Voicu, S. Wolff, and X. Yang, "The DYNES instrument: A description and overview," *Journal of Physics: Conference Series*, vol. 396, no. 4, p. 042065, 2012. [Online]. Available: http://stacks.iop.org/1742-6596/396/i=4/a=042065

[14] J. Zurawski, E. Boyd, T. Lehman, S. McKee, A. Mughal, H. Newman, P. Sheldon, S. Wolff, and X. Yang, "Scientific data movement enabled by the DYNES instrument," in *Proceedings of the First International Workshop on Network-aware Data Management*, ser. NDM '11. New York, NY, USA: ACM, 2011, pp. 41–48. [Online]. Available: http://doi.acm.org/10.1145/2110217.2110224